

Introducción:

En el presente documento se aprenderá a establecer rutas para vincular las diferentes páginas que conforman el sitio web. Además se creará una página de error para integrarla cuando un usuario intente acceder a un recurso inexistente.

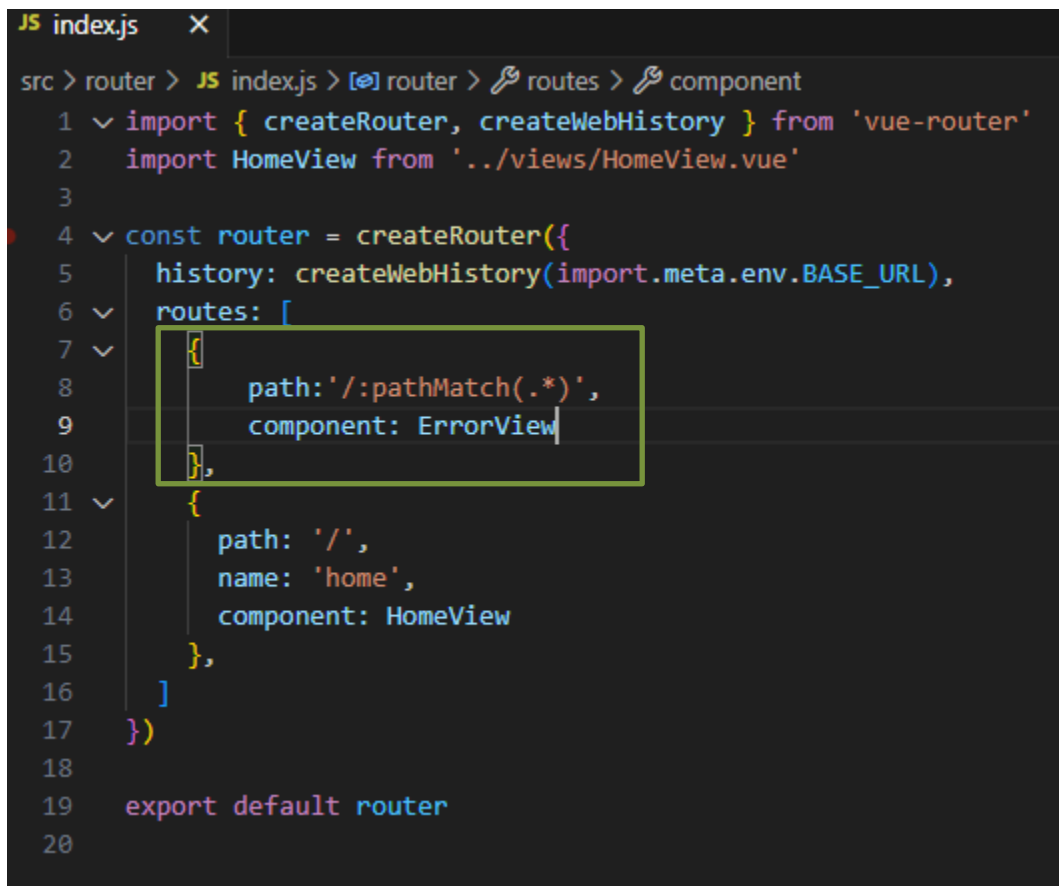
//-----

0.- Recursos necesarios:

- Un proyecto iniciado con vuejs y limpio de los archivos default.

Nota general: Para esta práctica nos centraremos en la carpeta src y sus recursos internos.

1. Ingresar al archivo index.js en la sección de router.
2. -Construiremos y agregaremos el código necesario para crear la ruta hacia la página de “not found” (página de error)



```
JS index.js X
src > router > JS index.js > router > routes > component
1  import { createRouter, createWebHistory } from 'vue-router'
2  import HomeView from '../views/HomeView.vue'
3
4  const router = createRouter({
5    history: createWebHistory(import.meta.env.BASE_URL),
6    routes: [
7      {
8        path: '/*:pathMatch(.*)',
9        component: ErrorView
10     },
11     {
12       path: '/',
13       name: 'home',
14       component: HomeView
15     },
16   ],
17 })
18
19 export default router
20
```

-Ahora agregaremos las rutas para las paginas que se incluirán en el contexto de este tutorial:

```
import { createRouter, createWebHistory } from 'vue-router'
import HomeView from '../views/HomeView.vue'

const router = createRouter({
  history: createWebHistory(import.meta.env.BASE_URL),
  routes: [
    {
      path: '/*pathMatch(.*)',
      component: ErrorView
    },
    {
      path: '/',
      name: 'home',
      component: HomeView
    },
    {
      path: '/login',
      name: 'login',
      component: LoginView
    },
    {
      path: '/singIn',
      name: 'singIn',
      component: SingInView
    }
  ]
})

export default router
```

-Ahora agregamos el import de los componentes para los que creamos las rutas:

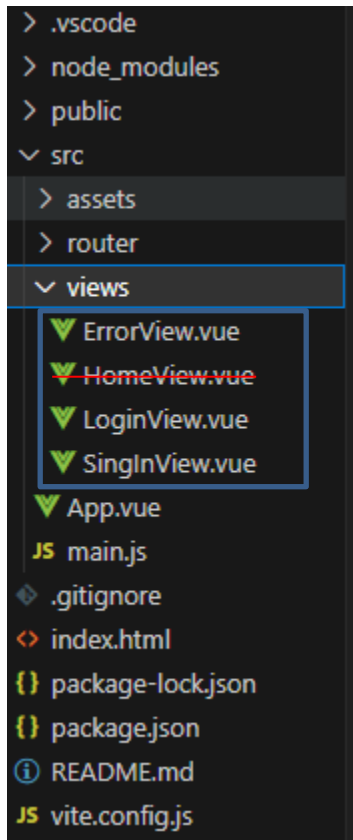


```
src > router > JS index.js > ...
1  import { createRouter, createWebHistory } from 'vue-router';
2  import HomeView from '../views/HomeView.vue';
3  import ErrorView from '../views/ErrorView.vue';
4  import LoginView from '../views/LoginView.vue';
5  import SingInView from '../views/SingInView.vue';
6
7  const router = createRouter({
8    history: createWebHistory(import.meta.env.BASE_URL),
9    routes: [
10     {
11       path: '/*',
12       component: ErrorView
13     },
14     {
15       path: '/',
16       name: 'home',
17       component: HomeView
18     },
19     {
20       path: '/login',
21       name: 'login',
22       component: LoginView
23     },
24     {
25       path: '/singIn',
26       name: 'singIn',
27       component: SingInView
28     },
29   ]
30 })
31
32 export default router
```

NOTA: Para este momento deberían salirte errores en el proyecto, debido a que todavía no se han creado las vistas para las cuales se agregaron las rutas.

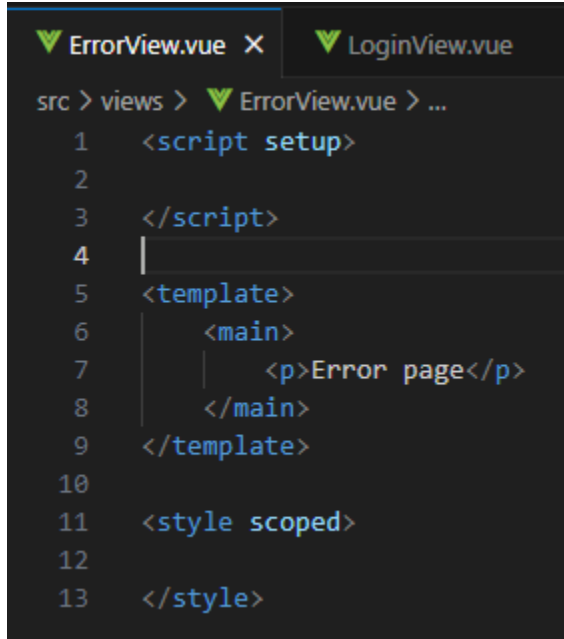
3. Ahora crearemos las vistas para las cuales creamos anteriormente las rutas, para este caso, la página de registro, de login y de error. La de HomeView recordemos que ya se encontraba ahí.

La extensión de archivo es *.vue



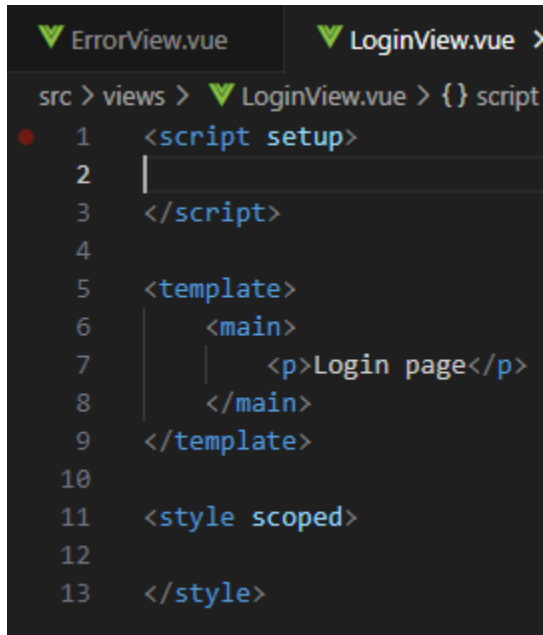
4. Una vez creados escribiremos el código básico necesario para el funcionamiento del ruteo entre paginas.

Para ErrorView:



```
src > views > ErrorView.vue > ...
1  <script setup>
2
3  </script>
4
5  <template>
6    <main>
7      <p>Error page</p>
8    </main>
9  </template>
10
11 <style scoped>
12
13 </style>
```

Para LoginView:



```
src > views > LoginView.vue > {} script
1  <script setup>
2
3  </script>
4
5  <template>
6    <main>
7      <p>Login page</p>
8    </main>
9  </template>
10
11 <style scoped>
12
13 </style>
```

Para SingIn

```
▼ ErrorView.vue  ▼ LoginView.vue  ▼ SingInView.vue X
src > views > ▼ SingInView.vue > {} script setup
1  <script setup>
2  |
3  </script>
4
5  <template>
6  |   <main>
7  |     <p>Sing in page</p>
8  |   </main>
9  </template>
10
11 <style scoped>
12
13 </style>
```

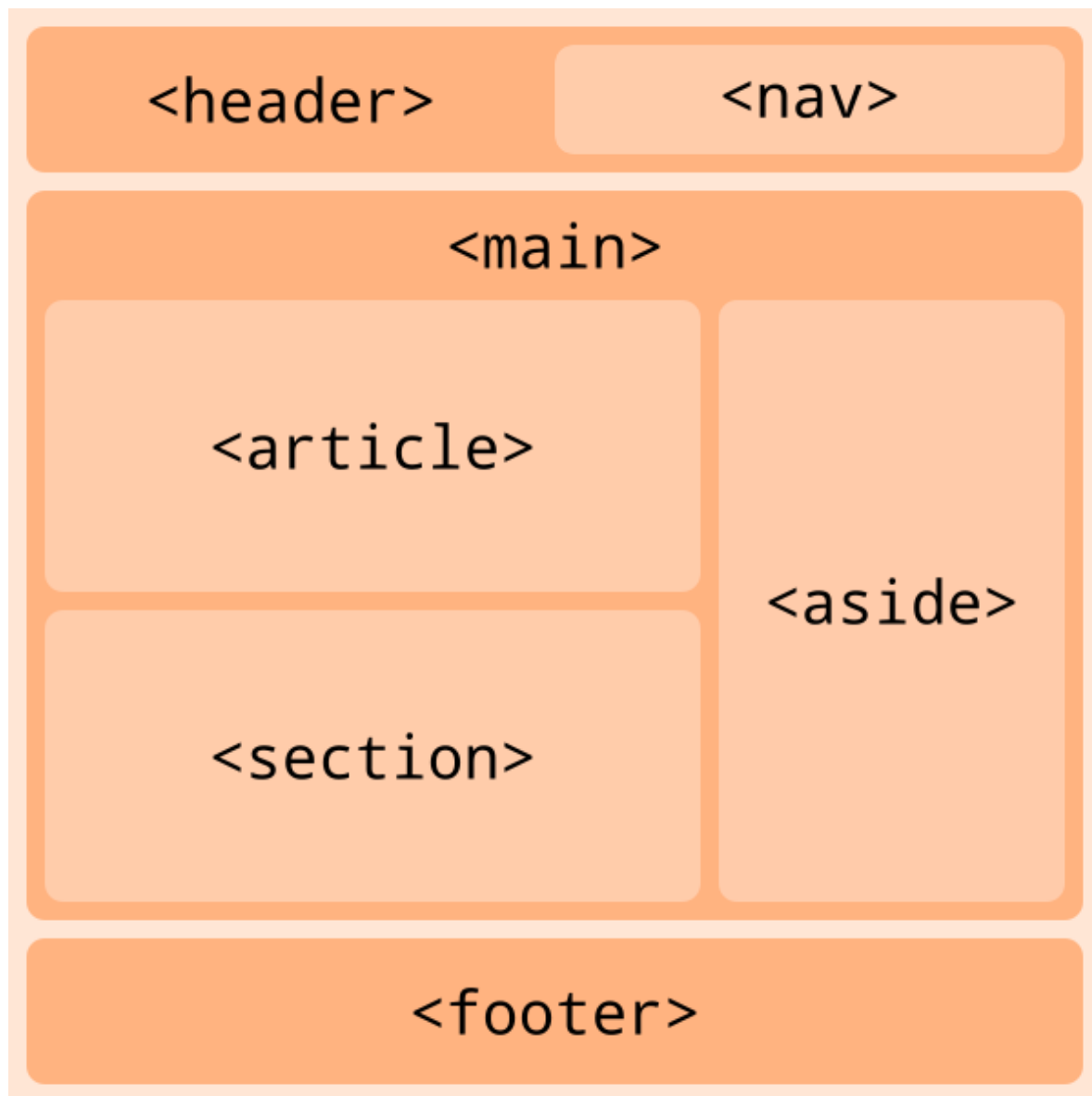
5. Configuraremos la página principal (HomeView) para poder dirigirnos hacia las otras paginas. Para este ejemplo usaremos la siguiente convención:

Esta es la estructura básica de un archivo de html

Estructura básica



Gracias a Vuejs, no necesitamos colocar como tal toda la estructura básica, podemos simplemente establecer la estructura interna del body (solo la necesaria)



Esto se vio en el paso 4.

- Ahora si procedemos a abrir el archivo HomeView y lo configuraremos de la siguiente manera:

```
▼ HomeView.vue X
src > views > ▼ HomeView.vue > {} script setup
1  <script setup>
2  | import TheHeader from '@components/TheHeader.vue';
3  </script>
4
5  <template>
6  | <main>
7  |   <TheHeader/>
8  | </main>
9  </template>
10
11 > <style scoped> ...
19 </style>
```

Cuadro azul: Importamos el archivo (aun no lo creamos) que constituirá el header general de la pagina

Cuadro morado: Llamamos el componente que acabamos de importar

-Crearemos el componente TheHeader.vue en su carpeta correspondiente:

```
▼ HomeView.vue  ▼ TheHeader.vue X  JS index.js
src > components > ▼ TheHeader.vue > {} style scoped
1  <script setup>
2  import { RouterLink } from 'vue-router';
3
4  </script>
5
6  <template>
7  |   <header>
8  |     <section class="logo">
9  |       <p>logo</p>
10 |     </section>
11 |     <nav>
12 |       <RouterLink to="/login">Iniciar sesion</RouterLink>
13 |       <RouterLink to="/singIn">Registrarse</RouterLink>
14 |     </nav>
15 |   </header>
16 </template>
17
18 <style scoped>
19 </style>
```


Aquí observamos que por medio de vue-router, en lugar de usar directamente la etiqueta <a>, utilizaremos <RouterLink> para crear el enlace a las otras páginas.

```
12 <RouterLink to="/login">Iniciar sesion</RouterLink>
13 <RouterLink to="/singIn">Registrarse</RouterLink>
```

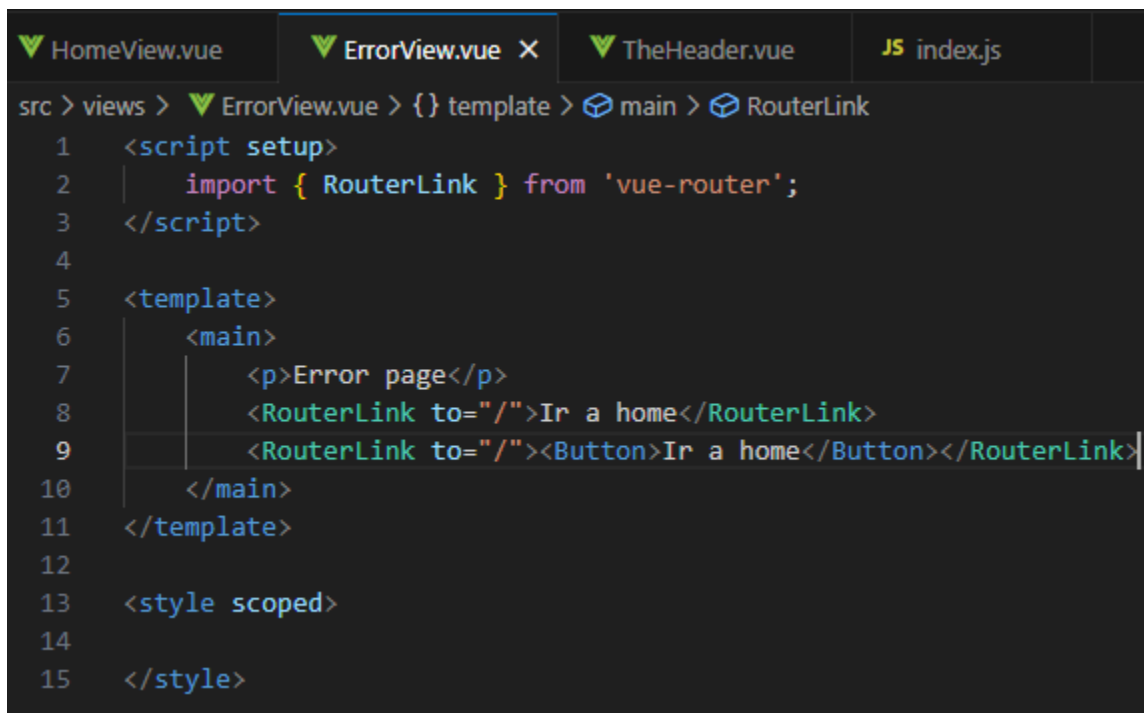
Identifica que lo que se coloca en la propiedad to= es la misma ruta que configuramos en el archivo index de rutas.

```
    path: '/login',
    name: 'login',
    component: LoginView
  },
  {
    path: '/singIn',
    name: 'singIn',
    component: SingInView
  },
]
```

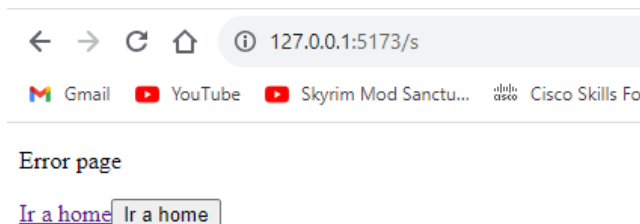
6. Si para este punto realizaste todo correctamente deberas visualizar algo como esto:



Recuerda que las rutas que has configurado se pueden utilizar en/y para diferentes elementos html, por ejemplo



Aquí se colocó en la página de error un link hacia el home y este mismo se agrego en un botón, así mismo, podría ser una imagen, un elemento de una lista, un elemento de menú, etc.



Nota final: Recuerda que no se le han colocado ningún tipo de estilos a las paginas, por eso es que se ven con esa apariencia.